

浪脚本零基础入门系列(十)

macroscript 编写快捷方式

作者: 飞浪

声明:本教程为 **CG++** 原创,转载请注明出处,谢谢合作:)

本节关键词:macroscript filein

不写教程很多个月了! 生活在继续, maxscript 也要继续。

macroscript 是 maxscript 中很重要的一部分,但是也很简单。想自己设置快捷键的朋友,不需要前面的章节的基础,直接看本节就知道怎么弄了。

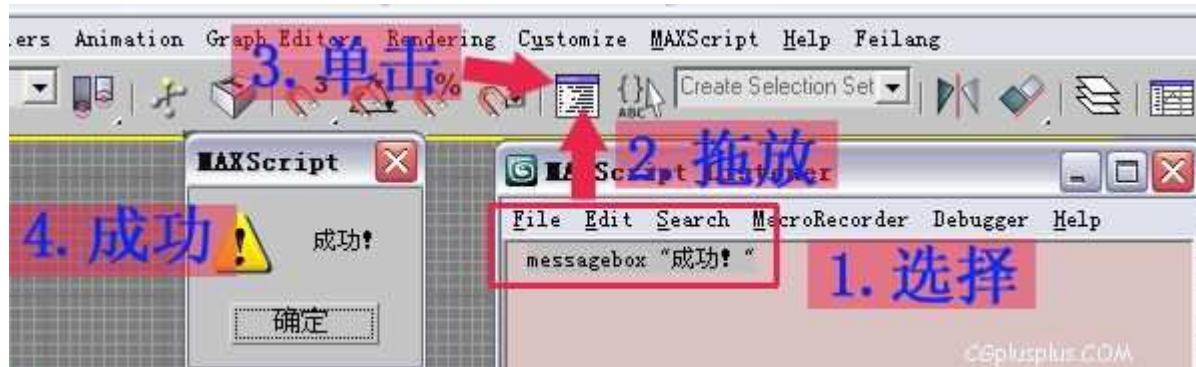
macroscript 宏脚本

macroscript 是一种生成工具栏按钮的脚本,当然也可以设置快捷键,右键菜单等等。下面来看个简单的实例:

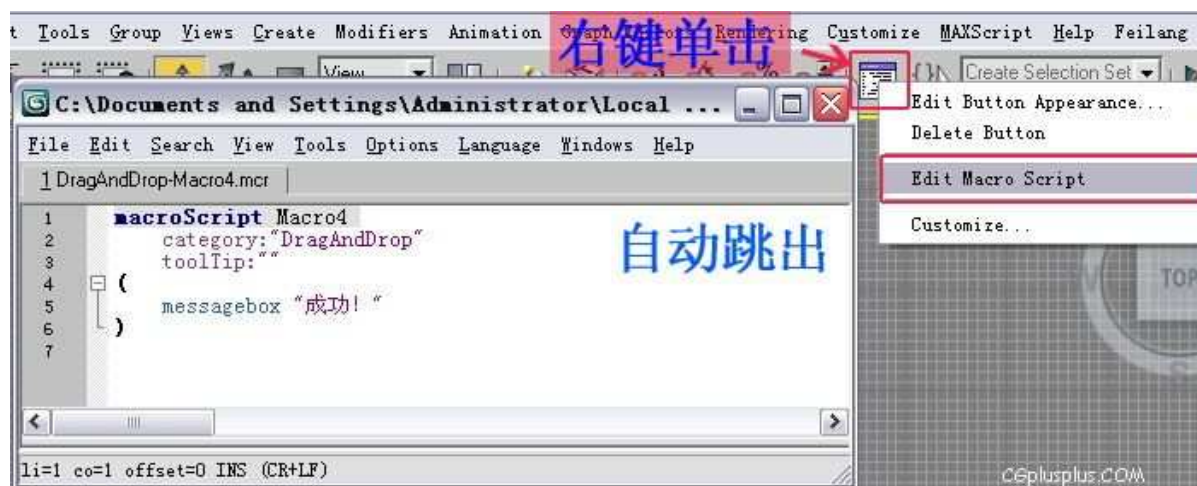
打开 MAXscript --> MAXscript listener, 在红色区域复制以下内容:

```
messagebox "成功!"
```

选中这些代码并拖放到工具栏上,这时候工具栏上自动生成了一个按钮。单击此按钮,如果你看到了如图所示的对话框“成功!”,那么恭喜你,你真的成功了。是不是很简单?



下面我们来看看真正的 macroscript 是怎么写的。在刚才生成的那个按钮是右键单击,选择并单击“Edit Macro Script”,这样就会自动跳出一个 maxscript 编辑器,这里才是真正的 macroscript 代码。



代码如下:

```
macroScript Macro4

category: "DragAndDrop"

tooltip: ""

(
    messageBox "成功! "
)
```

这个代码里面只有一句是我们刚才编写的, 即 `messageBox "成功!"`, 其他的代码均为 max 自动生成, 逐行解释如下:

macroScript Macro4 --前面 `macroScript` 是申明, 表示这是一个 `macroscript` 脚本, 后面的 `Macro4` 是变量, 可以自行更改, 这个变量是自动生成, 可能你的并不是 `Macro4`, 而是 `Macro1, Macro2...`

category: "DragAndDrop" --`category` 表示目录, 设置快捷键的时候经常会遇到, "`DragAndDrop`"是目录的名字, 可以是任意字符串, `DragAndDrop` 的字面意思是拖放。

tooltip: ""--表示按钮提示, 后面的双引号里面没有字符, 就没有提示了, 可以改成 `tooltip: "这是一个按钮"`。

后面有一对括号, 所有要执行的代码都扔到这个括号里面。下面来看看 `macroscript` 究竟怎么来调用。

依次打开 `Customize -> Customize User Interface --> 选择 Toolbars` 看看这个面板。Group 默认是 `MainUI`, Category 默认是 `All Commands` 现在把 `All Commands` 下拉后选择 `DragAndDrop`, 下面就有我们刚才的脚本, 我的是 `Macro4`, 在它上面点右键, 可以选择编辑此脚本。好, 我们来编辑一下脚本改成如下:

```
macroScript Macro4
```

```
category:"DragAndDrop"

toolTip:"这是我的按钮"

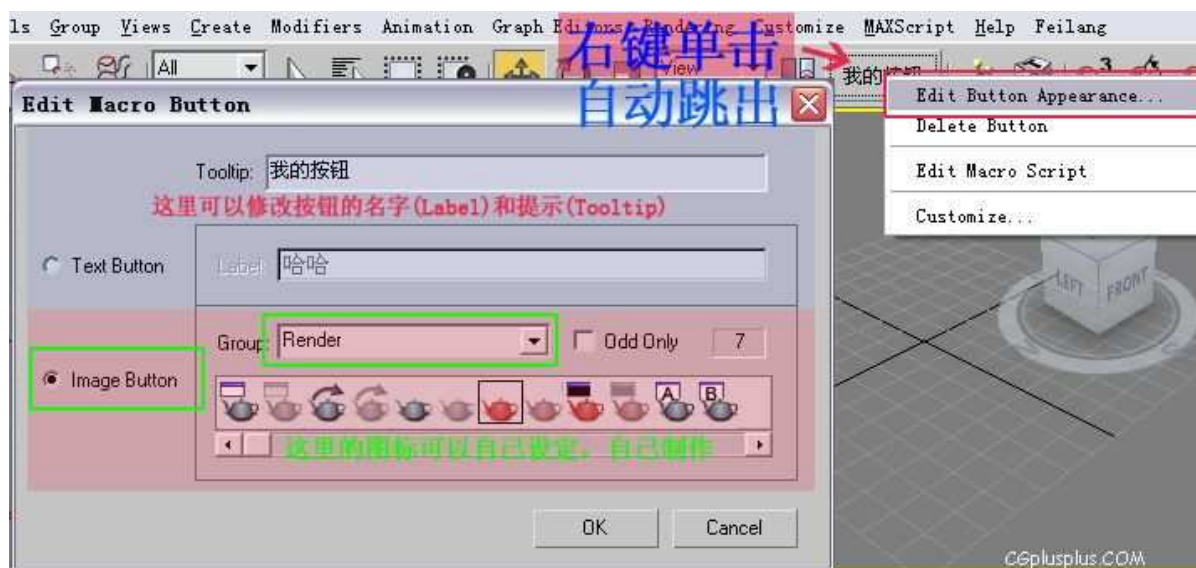
buttontext:"我的按钮"

(
messagebox "成功! "
)
```

这里加了一行 **buttontext:"我的按钮"**，修改之后按快捷键 **ctrl+E** 执行脚本，再关闭并重新打开一下 **Customize User Interface**，这时 **Macro4** 已经变成了“我的按钮”，没错，这个 **buttontext** 就是此脚本的显示文字，把它拖放到工具栏上，按钮的名字会是“我的按钮”。

制作按钮的图标

在刚才制作的按钮上面单击右键，选择“**Edit Button Apperance**”，可以修改按钮的显示方式。如图所示，上面的 **Text Button** 可以修改按钮的名字和提示语。我们着重来讲一下下面的 **Image Button**。选择 **Image Button**，在 **Group** 后面选择 **Render**，我们会看到下面有一排图标，你可以选择一个图标单击 **OK**，那么刚才那么按钮的图标就变成了你选择的图标。



如何自己制作这样的图标呢？现在在 **max** 根目录下打开以下路径：...\\Autodesk\\3ds Max 2009\\ui\\Icons，找到文件 **Render_16a.bmp**，按名称排列的话依次会看到：**Render_16i.bmp**，**Render_24a.bmp**，**Render_24i.bmp**，打开这些图片一看，原来这就是刚才我们设置按钮图标时候看到的图标啊。制作这样的图标很简单，只要遵循以下几条即可：

1、图标大小分两种，如，**24X24**，和 **16X15**，其中图标的高度不能变，高度是

24 的图片，宽度必须是 24 的整数倍，高度是 15 的图片，宽度必须是 16 的整数倍。两个图标的倍数要相同。即图标大小应该为： $(N*24)X24$ 和 $(N*16)X15$ 。

2、一组图标必须有 4 个，其命名规则是，使用统一的前缀，后缀名字依次是 `_16a.bmp`, `_16i.bmp`, `_24a.bmp`, `_24i.bmp`，如上面的 Render 系列图标，以 `_16i.bmp`, `_24i.bmp` 结尾的图标是最终显示在 max 界面上的图标，以 `_16a.bmp`, `_24a.bmp` 结尾的图标是黑白通道，控制图标需要显示的部分。

3、图标文件必须放到 max 根目录的 `ui\Icons` 文件夹下面。刚放进去的图标，重启 MAX 才生效。

本文最后附上场景助手的图标，请大家先下载，下面要拿来测试。

图标做好了，怎么写进 `macroscript` 呢？看如下代码：

```
macroScript Macro4
category: "DragAndDrop"
toolTip: "我的按钮"
buttonText: "我的按钮"
icon: #("FeilangTools", 3)
(
messageBox "成功！"
)
```

代码 `icon: #("FeilangTools", 3)` 即为调用图标，因为我制作的图标文件前缀名是 `FeilangTools`，这里调用就写这个名字，后面的 3 表示图标的第中的三个，我一共做了三个图标并列，所以大小分别是 `72X24` 和 `48X15`。执行这个代码，再重新打开 **Customize User Interface** 找到 **Toolbars**，按上面讲解选择 **DragAndDrop**，可以看到刚才编写的 `macroscript` 已经有了图标了，这时再拖到工具栏上就会显示自己制作的图标。

上面一直讲的是 `macroscript` 作为快捷方式的编写，下面着重来看看那个括号里面的代码。把代码

```
messageBox "成功！"
```

用记事本保存到 E 盘下，格式改为 `.ms`，再修改 `macroscript` 那个括号里面的代码如下：

```
macroScript Macro4
category: "DragAndDrop"
toolTip: "我的按钮"
```

```
buttontext:"我的按钮"

icon:("#FeilangTools",3)

(
filein "E:\\myscript.ms"
)
```

执行此代码，重新设置按钮到工具栏，单击按钮，如果你看到了对话框“成功！”，那么恭喜你，你又真的成功了^_^！

filein "脚本路径" 可以执行其他的脚本，用在这里你可以调用任何其他的脚本了，支持 **.ms** , **.mse** , **.mcr** (用 **filein** 调用这个 **.mcr** 文件没有意义，这里只是列出来)。再来看：

```
macroScript Macro4

category:"DragAndDrop"

toolTip:"我的按钮"

buttontext:"我的按钮"

icon:("#FeilangTools",3)

(
on execute do
( s=sphere()
s.radius = 100
)
)
```

on execute do 是当 **macroscript** 被“召唤”的时候，需要运行的代码，运行此代码，单击工具栏按钮，此脚本会创建一个球体，半径为 **100**。**on execute do** 后面的括号可以是任意的代码块（**block**）。继续修改代码，只修改 **macroscript** 括号部分：

```
(
on execute do
( s=sphere()
s.radius = 100
)
on isChecked do selection.count==1
)
```

当你按下工具栏的按钮时，按钮状态会变吧，这个 `on isChecked do` 就可以控制按钮状态。这句后面可以是任意代码块，但是必须返回 `true` 或 `false`。如果返回 `true`，按钮会自动“被按下”，如果返回 `false`，按钮会弹起来。运行刚才修改的 `macroscript`，当你选择一个物体的时候，按钮就“被按下”，如果不选择物体或者选择多余一个物体，按钮就“被弹起”，（最近流行被字句...）

`macroscript` 里面还有其他一些事件，大概说一下，有兴趣的可以看看：

`on isEnabled do <expr> --<expr>` 返回 `true` 或 `false`。当它是 `false` 的时候，如果你设置的快捷方式是右键菜单或者菜单栏菜单或按钮，则不可用（灰色显示）如：

`on isEnabled do selection.count!=0`

`on isVisible do <expr> --<expr>` 返回 `true` 或 `false`。当它是 `false` 的时候，如果你设置的快捷方式是右键菜单或者菜单栏菜单，则看不到此脚本。如果是按钮，没影响，如：

`on isVisible do selection.count!=0`

`on closeDialogs do <expr> --`这句是前提是必须要有 `on isChecked do` 语句，`isChecked` 后面返回 `false` 时，`on closeDialogs do` 将代替 `on execute do` 语句。这样，可以用按钮等快捷方式控制在 `macroscript` 里面的 `rollout` 的开和关。例如：

```
macroScript aNewTest
category:"CGplusplus"
toolTip:"macroscript 测试"
buttonText:"macroscript 测试"
icon:$("#FeilangTools",1)
(
rollout test "test" width:248 height:169
(
label lbl "欢迎常回来看看！我们的网址是：" pos:[20,40] width:225 height:13
HyperLink hy "www.cgplusplus.com" pos:[65,67] width:100 height:15 color:(color 0
255 0) hovercolor:(color 255 127.5 0) visitedcolor:(color 0 255 0)
address:"http://www.cgplusplus.com/"
)
on isChecked do test.open
on execute do
( createdialog test bgcolor:black fgcolor:red
)
on closeDialogs do
```

```
( destroydialog test
)
)
```

运行此脚本，设置成按钮，并单击。如果你能看到对话框里面写着：

欢迎常回来看看！我们的网址是：www.cgplusplus.com

那么恭喜你，你真的学会了^_^！那本人也该退场了！哦，还加一句：macroscript 不能加密，不信你试试^_^（加密方式是 encryptscript "E:\\haha.mcr"）。

本节到此结束，谢谢捧场^_^

场景助手图标：



本节原帖地址：<http://www.cgplusplus.com/bbs/viewthread.php?tid=902>

本节结束。